Peer-based query rewriting in SPARQL for semantic integration of Linked Data

Mirko Michele Dimartino

Birkbeck, University of London
mirko@dcs.bbk.ac.uk

Abstract. In this proposal we address the problem of ontology-based SPARQL query answering over distributed Linked Data sources, where the ontology is given by conjunctive mappings between the source schemas in a peer-to-peer fashion and by equality constraints between constants. In our setting, the data is not materialised in a single datastore: it is accessed in a distributed environment through SPARQL endpoints. We aim to achieve query answering by generating the *perfect rewriting* of the original query and then processing the rewritten query over distributed SPARQL endpoints. We identify a subset of ontology constraints that enjoy the *first-order rewritability* property and we perform preliminary empirical evaluation taking into account such restricted constraints only. For future work, we aim to tackle the query answering problem in the general case.

1 Scene Setting

The Web of Linked Open Data (LOD) has developed from a few datasets in 2007 into a large data space containing billions of RDF triples published and stored in hundreds of independent datasets. This huge information cloud, ranging over a wide set of data domains, poses a great challenge when it comes to reconciling heterogeneous schemas or vocabularies adopted by data publishers. According to Linked Data best practices [10], data publishers should reuse terms from widely-used vocabularies already present in the cloud, in order to enable the discovery of additional data and to support the integration of data from multiple sources. However, commonly-used vocabularies usually do not provide all terms needed to completely describe the content of the data. Thus, data providers often define proprietary terms as sets of new IRIs published on the cloud. This trend leads to the formation of islands of data describing overlapping domains, rather than generating a global knowledge base.

Over the past years, researchers in the Semantic Web community have attempted to tackle these challenges by proposing several approaches based on semantics preserving SPARQL rewriting algorithms. These methods allow users to pose SPARQL queries expressed adopting a preferred vocabulary and a rewriting algorithm provides translations of the query in the language of similar vocabularies. To rewrite queries, they reason over semantic mappings between the sources. Following this, the rewritten query is evaluated over the sources and a more complete answer is returned to the user. Several approaches in the literature address this problem, however most of these works are based on the common two-tiered local-to-global schema integration paradigm, where

query are expressed over the global schema and are reformulated in the language of the source vocabulary, to be then evaluated over the data stored at the source. By contrast, in the Linked Data cloud each data store is an autonomous system whose vocabulary should represent part of the global schema, available from a distributed environment. In this regard, Linked Data consumers should be able to pose queries adopting any of the source vocabularies and to access similar sources through query translation, in a transparent way and without relying on a single global schema. Following this idea, we believe that a peer-to-peer approach is more suitable than a local-to-global approach because it provides a more decentralised architecture where peers act both as clients and as servers during the query reformulation process. In addition, the local-to-global approaches typically require a comprehensive global schema design before they can be used, thus they are difficult to scale because schema evolution may break backwards compatibility. Scalability is a key property of LOD-oriented data mediation systems, due to the continuous increase of data published on the web.

In this proposal paper, we address the problem of ontology-based SPARQL query answering via query rewriting over Linked Data sources, where the ontology is given by conjunctive mappings between the source schemas in a peer-to-peer fashion, and by equality constraints between constants for entity resolution. We focus on a setting where data is not materialised in a single datastore; it is accessed through distributed SPARQL endpoints. Query answering is achieved via: *(i)* computing the *perfect rewriting* with respect to the ontology; the original query is reformulated so as to obtain a sound and complete answer based both on the extensional database (i.e., the stored RDF triples) and the ontology defined as the set of data constraints entailed by the mapping assertions; *(ii)* processing the perfect rewriting of the original query over distributed SPARQL endpoints. Differently from other approaches which focus on tractable mapping languages, we aim to tackle query answering based on conjunctive mappings, i.e., positive rules in which the conclusion may contain existentially quantified variables, which makes reasoning tasks undecidable if interpreted in first order semantics. In addition, we take into account the distributed nature of the LOD cloud by processing the queries directly over the sources through their public SPARQL endpoints, without the need to materialise the RDF sources in a centralised middleware.

This PhD thesis aims to address the following research questions:
RQ1 How can we achieve ontology-based SPARQL query answering where the ontology comprises conjunctive mappings, i.e., existential rules which lead to undecidability of reasoning tasks?
RQ2 How can we compute the perfect rewriting of conjunctive SPARQL queries with respect to conjunctive mappings (interpreted so as to preserve decidabilty) and equality constraints between RDF sources? Which query language is suitable for such rewritings?
RQ3 How do we process the rewritten query taking into account that the RDF sources are not materialised is a single datastore, and data is accessed via distributed SPARQL endpoints?

Related work. Several works in the literature address data mediation for Linked Data. Very close to our work is [5] which proposes an algorithm to rewrite SPARQL queries in order to achieve integration of RDF databases. The approach is based on

the encoding of rewriting rules for RDF patterns that constitute part of the structure of a SPARQL query. The adopted rules, called *Entity Alignments*, express semantic mappings between two datasets and can be interpreted as definite Horn clauses in First-Order (FO) logic where only the *triple* predicate is used. It is then based on the well-known *global-as-view* [13] data integration approach, where a term of the global schema is mapped to a view of the source. The main limitation of the proposed approach is that it is not applicable when a more expressive formalism is needed to align two schemas, for example, when the relations in the sources need to be specified as views over the mediated schema. In this scenario, the expressive power of the *local-as-view* [13] formalization is also necessary. One interesting aspect is that the framework deals with co-reference resolution by including *Functional Dependencies* in the mapping rules. Similar limitations are in the approach proposed by Makris et al. [15, 16] which uses a mapping language based on Description Logics, defining 1:N cardinality mapping types where a term from one vocabulary is mapped to a Description Logic expression over another vocabulary. Other similar approaches leveraging less expressive formalisms for data mediation can be found in [17, 20, 21]. For instance, [17] proposes SemLAV, an alternative technique to process SPARQL queries without generating rewritings. SemLAV executes the query against a partial instance of the global schema which is built on-the-fly with data from the relevant views. Work in [20] addresses rewriting techniques that consider only co-reference resolution in the rewriting process, and [21] adopts a small set of mapping axioms defined only by those RDF triples whose predicate is one of the following OWL or RDFS terms: `sameAs`, `subClassOf`, `subPropertyOf`, `equivalentClass`, and `equivalentProperty`. Other similar approaches are proposed in [12, 14, 19]. All the above-mentioned frameworks address query answering over two-tiered architectures and tractable mapping languages, while we wish to explore more general settings.

Several peer-to-peer systems for RDF datasources can be found in the literature. For instance, in [2, 3] the authors describe a distributed RDF metadata storage, querying and subscription service, as a structured P2P network. Similarly, work in [18] proposes routing strategies for RDF-based P2P networks. These are non-database-oriented tools that have little support for semantic integration of highly heterogeneous data. In fact, they focus strictly on handling semantic-free requests which limits their utility in establishing complex links between peers.

2 Proposed Approach

Our approach to semantic integration of heterogeneous Linked Data sources is based on the *RDF Peer System* (RPS) introduced in our recent paper [7]. This is a framework for peer-based integration of RDF datasets, where the semantic relationships between data at different peers are expressed through mappings. Formally, an RPS $\mathcal{P}$ is defined as a tuple $\mathcal{P} = (\mathcal{S}, G, E)$, where $\mathcal{S}$ is the set of the *peer schemas* in $\mathcal{P}$, G is a set of *graph mapping assertions* and E is a set of *equivalence mappings*. A peer schema in $\mathcal{S}$ represents the adopted vocabulary, that is, the set of IRIs that a peer (i.e. an RDF data source) adopts to describe its data. The sets of schema-level mappings and instance-level mappings between peers are given by G and E, respectively. G provides semantic linkage

between the schemas of different peers and contains mapping assertions of the form $Q \rightsquigarrow Q'$, where Q and Q' are conjunctive SPARQL queries with the same arity over two peers, e.g.: $q(x, y) \leftarrow (x, actor, y) \rightsquigarrow q(x, y) \leftarrow (x, starring, z)$ AND $(z, artist, y)$, where the query $q(x, y) \leftarrow (x, pred, y)$ evaluated over an RDF database returns the subjects and objects appearing on all the triples whose predicate is $pred$. For instance, this mapping assertion states that, if there is a triple in the first source of the form $(IRI_1, actor, IRI_2)$, then the two triples $(IRI_1, starring, \_{:}b)$ and $(\_{:}b, artist, IRI_2)$ need to be exported to the other source, where $\_{:}b$ is a blank node. Mappings in E are of the form $c \equiv_e c'$, where c and c' are IRIs located in the same peer or in two different peers. Equivalence mappings are used to solve the problem of identity in the Semantic Web scenario. In fact, an IRI ensures to uniquely identify a resource on the web, not the entity the resource represents [9]. To partially cope with this, LOD publishers often use the built-in OWL property `sameAs`[1], to explicitly "align" the newly created IRIs with existing IRIs that represent the same real-world entity. Equivalence mappings entail the semantics of `sameAs`.

Query processing in our setting is performed by query rewriting; the original query is reformulated so as to obtain a sound and complete answer based both on the extensional database (i.e., the stored RDF triples) and the ontology defined as the set of data constraints entailed by the mapping assertions. Different from the existing SPARQL rewriting approaches, our integration framework preserves the expressive power of both the global-as-view and local-as-view integration formalisms. In addition, we adopt a more general network of interrelated peer-to-peer relations. To the best of our knowledge, none of the existing approaches addresses SPARQL query answering under such a setting.

For the theoretical evaluations, we formalise the query answering problem by generalising the notion of *certain answers* [1] to our context. We show that the problem is subsumed by conjunctive query answering in data exchange for the relational model, and that a conjunctive SPARQL query can be answered in polynomial time in data complexity [7]. Decidability is preserved since only the certain answers are propagated through conjunctive peer mappings (see [7] for more details). We argue that this is an advantage of our approach, since an arbitrary interconnection of conjunctive peer mappings leads to undecidability if interpreted in FO semantics. Although they preserve decidability, we show that our peer mappings define non-FO-rewritable constraints, and so it is not possible to process queries in general RPSs by rewriting them into SPARQL queries. In this regard, for the preliminary empirical evaluation, we implement a SPARQL rewriting algorithm that leverages only FO-rewritable sets of peer mappings. As future work, we aim to investigate rewriting algorithms that produce queries in a language more expressive than FO-queries, in order to implement the full semantics of our system.

3 Implementation of the Proposed Approach

This section illustrates the main components of a middleware for LOD integration, which is also proposed in our recent paper [6]. The system provides a query inter-

[1] http://sameas.org

face between the user and the Linked Data sources and it is based on the formalisation of the RPS illustrated in the previous section. To summarise, our middleware exposes a unified view of heterogeneous RDF sources which are semantically linked with the RPS mapping assertions. A unified SPARQL endpoint accepts queries expressed in any source vocabulary. A SPARQL query rewriting engine rewrites the queries with respect to the semantic mappings of an instance of RPS, so as to retrieve more complete answers. Then, the rewritten query is evaluated over the sources in a federated approach and the query result is presented to the user.

In our system, the query rewriting engine is composed of two sub-engines. *(i)* The *semantic integration* module generates a "perfect rewriting" of the user's query, that is, a query that returns, once evaluated, a sound and complete answer of the original query based on the semantic mappings in the RPS. *(ii)* The *query federation* module executes a second rewriting step adopting the SPARQL 1.1 extension and exploiting the `SERVICE` clause; it generates a federated query to be evaluated over multiple RDF sources.

The system provides for *automated alignment* of the peer schemas, to link entities and concepts in the Linked Open Data cloud. It extracts structural information from the sources, such as the sets of entities, predicates, classes etc. Then, it performs schema alignment and coreference resolution by: *(i)* retrieving mappings between sources, such as `owl:sameAs` or `VoID`[2] triples, and other semantic links between sources; *(ii)* generating new mappings, using existing ontology matching and instance linkage techniques, such as *Falcon-AO* [11]; *(iii)* translating these alignments into our peer mapping language; and *(iv)* storing the mappings in the RPS.

For a preliminary empirical evaluation, we implemented a partial version of system that leverages only FO-rewritable sets of peer mappings which are manually designed. For this setting, we develop a SPARQL rewriting algorithm, called RPS-rewrite, which is based on a backward chaining mechanism [8]. It takes as input a conjunctive SPARQL query and it generates the FO-rewriting of the input query with respect to an instance of RPS, as a union of conjunctive SPARQL queries. Furthermore, we address two query optimisations of the query resulting from RPS-rewrite. The first optimisation performs a pruning of all the SPARQL disjuncts with triple patterns that cannot provide a successful graph pattern match. The second optimisation is given by "ignoring" the equivalence mappings during the backward chaining steps, since they lead to a production of SPARQL disjuncts that grow exponentially with respect to the number of mapping assertions. Consequently, equivalence mappings are treated as stored `sameAs` triples and leveraged on query evaluation for co-reference resolution, by adopting a technique of variable rewriting which we omit for space reasons. These `sameAs` triples are stored externally on a *Virtuoso* server and are accessed through a SPARQL endpoint.

Regarding query federation, triple patterns in the body of the query are then grouped with respect to the RDF sources that can provide a successful graph pattern match. Then, the groups are assigned to the endpoints of the related sources, and evaluated using the SPARQL 1.1 `SERVICE` clause. Finally, the results are presented to the user.

[2] http://www.w3.org/TR/void/

4 Empirical Evaluation Methodology

The goal of the preliminary evaluation is to provide a study of the behaviour of the current version of the framework with the aim of *(i)* ensuring that our framework can be used in its restricted version and, *(ii)* analysing basic performance in terms of cost execution time, and *(iii)* detecting current weaknesses of our framework to suggest future developments. We select three large-scale datastores with overlapping vocabularies in the domain of movies: *DBpedia*, *Linked Movie Database* and *Fact Forge*. The current version of our middleware is a *Java* application that takes as input a SPARQL query, generates the rewriting in SPARQL 1.1 and executes the rewritten federated query over the selected datastores using *Apache Jena*, the well-known open source Semantic Web framework for Java.

We performed a partial semantic alignment of DBpedia, Linked Movie Database and Fact Forge schemas, defining a set of FO-rewritable one-to-one mappings for similar classes and predicates, adopting the RPS mapping language. For instance, we define a mapping of the form:
$q(x, y) \leftarrow (x, \texttt{linkedmdb:actor}, y) \rightsquigarrow q(x, y) \leftarrow (x, \texttt{dbpedia:starring}, y)$
to express a 1:1 *predicate mapping* from the IRI `linkedmdb:actor` to the IRI `dbpedia:starring` , and, a mapping of the form:
$q(x) \leftarrow (x, \texttt{rdf:type}, \texttt{ff:Person}) \rightsquigarrow q(x) \leftarrow (x, \texttt{rdf:type}, \texttt{foaf:Person})$
to express a 1:1 *class mapping* from the IRI `ff:Person` to the IRI `foaf:Person`, leveraging the semantics of the built-in RDF predicate `rdf:type` for the class mappings. Also, we retrieved some `sameAs` triples from the sources and we generated new triples so as to encode the reflexive, symmetric and transitive closure of the `sameAs` binary relation; this provides co-reference resolution of IRIs as we explained in the previous section. The peer mappings obtained present arbitrary topologies and include some mapping cycles.

To conduct our tests, we generate a set of SPARQL queries with up to three triple patterns in the body. We then evaluate the queries over the three endpoints of the datastores in order to obtain our baselines. Finally we execute the queries on our middleware, and we compare the number of results retrieved and the query execution time. The results are shown in Figure 1 and allow us to derive two main insights. As expected, the amount of information retrieved increases significantly by adopting our system, due to its interoperability with heterogeneous vocabularies. In addition, the approach does not compromise query execution time, since overall the response time of our system can be seen as an average of the query response time over the single datastores. In fact, using the RPS can sometimes be faster than using just one single source endpoint. This may be due to the minimisation of the number of distributed-joins performed by Jena, which may increase the throughput with respect to the fastest endpoint (in our case DBpedia).

5 Lessons Learned, Open Issues, and Future Directions

In this paper we address the problem of integrating RDF data sources in a peer-based fashion, where mappings are defined between arbitrary peers, without a centralised

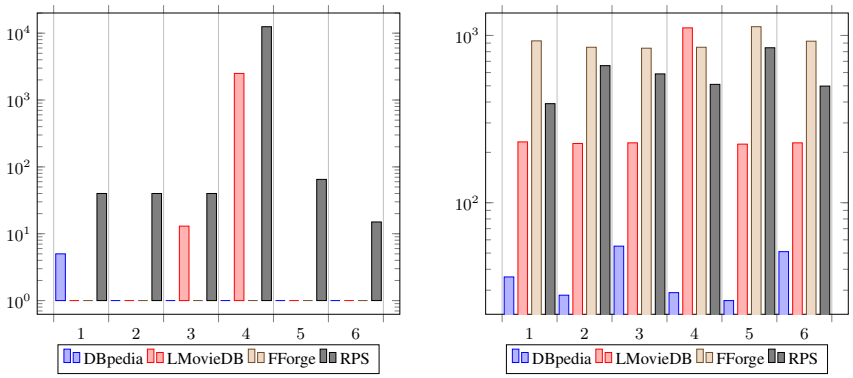

Fig. 1. Number of results on the left and query execution time on the right (logarithmic scales). Queries 1 - 6 shown on the x axes.

schema. We have proposed a novel formalisation of the notion of a peer-to-peer semantic integration system based on the Linked Open Data scenario. Following that, we have shown that query answering can be done in polynomial time in data complexity, and so we address Research Question RQ1 of this proposal. Finally, we have seen that it is not possible to process queries in general RDF peer systems by rewriting them into FO-queries, so we have conducted a preliminary empirical evaluation on FO-rewritable ontologies.

To address Research Question RQ2, we plan to devise a query rewriting algorithm that exploits the full semantics of our system. Firstly, we intend to investigate the possibility of adopting a *combined* approach, where the sources are partially materialised and queries are rewritten according to some of the dependencies only. To compute the perfect rewriting, another possible approach is to devise a rewriting algorithm that produces rewritten queries in a language more expressive than FO-queries, for instance Datalog, similarly to the approach in [4] which leverages new semantics for peer-to-peer systems based on epistemic logic. Another possible target language for the rewriting algorithm is SPARQL 1.1 with property paths; the idea is to leverage the expressive power of regular path queries in order to catch non-FO-rewritable constraints, such as the transitive closure of a relations. Two hypotheses follow from this approach: *(a)* it is possible to generate a SPARQL 1.1 query as a perfect rewriting of a conjunctive SPARQL query with respect to an RPS; *(b)* SPARQL 1.1 is not expressive enough: in this case we aim to characterise subsets of RPS mappings that are rewritable in SPARQL 1.1 with property paths.

Following from this, we will address Research Question RQ3. If the target language is SPARQL 1.1, query evaluation over multiple SPARQL endpoints can be done straightforwardly by exploiting the SPARQL 1.1 `SERVICE` clause. For Datalog rewritings, we will tackle the problem of distributed Datalog query processing on top of SPARQL endpoints.

Acknowledgments. I wish to thank my supervisors, Dr. Andrea Calì, Prof. Alexandra Poulovassilis and Dr. Peter Wood, for their invaluable support. I am also grateful to the reviewers for their constructive feedback.

References

1. Abiteboul, S., Duschka, O.M.: Complexity of answering queries using materialized views. In: Proc. of PODS. pp. 254–263 (1998)
2. Cai, M., Frank, M.: RDFPeers: a scalable distributed RDF repository based on a structured peer-to-peer network. In: Proc. of WWW. pp. 650–657 (2004)
3. Cai, M., Frank, M., Yan, B., MacGregor, R.: A subscribable peer-to-peer RDF repository for distributed metadata management. Web Semantics 2(2), 109–130 (2004)
4. Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Logical foundations of peer-to-peer data integration. In: Proc. of PODS. pp. 241–251 (2004)
5. Correndo, G., Salvadores, M., Millard, I., Glaser, H., Shadbolt, N.: SPARQL query rewriting for implementing data integration over Linked Data. In: Proc. of EDBT/ICDT Wksp (2010)
6. Dimartino, M.M., Calì, A., Poulovassilis, A., Wood, P.T.: Implementing peer-to-peer semantic integration of Linked Data. In: Proc. of BICOD (2015)
7. Dimartino, M.M., Calì, A., Poulovassilis, A., Wood, P.T.: Peer-to-peer semantic integration of Linked Data. In: Proc. of EDBT/ICDT Workshops. pp. 213–220 (2015)
8. Gottlob, G., Orsi, G., Pieris, A.: Ontological queries: Rewriting and optimization. In: Proc. of ICDE. pp. 2–13 (2011)
9. Halpin, H.: Identity, reference, and meaning on the web. In: Proc. of WWW Workshops (2006)
10. Heath, T., Bizer, C.: Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology 1(1), 1–136 (2011)
11. Hu, W., Qu, Y., Cheng, G.: Matching large ontologies: A divide-and-conquer approach. Data & Knowledge Engineering 67(1), 140–160 (2008)
12. Le, W., Duan, S., Kementsietsidis, A., Li, F., Wang, M.: Rewriting queries on SPARQL views. In: Proc. of WWW. pp. 655–664 (2011)
13. Lenzerini, M.: Data integration: A theoretical perspective. In: Proc. of PODS. pp. 233–246 (2002)
14. Lopes, F.L.R., Sacramento, E.R., Lóscio, B.F.: Using heterogeneous mappings for rewriting SPARQL queries. In: DEXA Workshops. pp. 267–271 (2012)
15. Makris, K., Bikakis, N., Gioldasis, N., Christodoulakis, S.: SPARQL-RW: transparent query access over mapped RDF data sources. In: Proc. of EDBT. pp. 610–613 (2012)
16. Makris, K., Gioldasis, N., Bikakis, N., Christodoulakis, S.: Ontology mapping and SPARQL rewriting for querying federated RDF data sources. OTM pp. 1108–1117 (2010)
17. Montoya, G., Ibáñez, L.D., Skaf-Molli, H., Molli, P., Vidal, M.E.: SemLAV: local-as-view mediation for SPARQL queries. TLDKS Journal XIII pp. 33–58 (2014)
18. Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Löser, A.: Super-peer-based routing strategies for RDF-based peer-to-peer networks. Web Semantics: Science, Services and Agents on the World Wide Web 1(2), 177–186 (2004)
19. Schenner, G., Bischof, S., Polleres, A., Steyskal, S.: Integrating distributed configurations with RDFS and SPARQL. In: 16th International Configuration Workshop. p. 9 (2014)
20. Schlegel, K., Stegmaier, F., Bayerl, S., Granitzer, M., Kosch, H.: Balloon fusion: SPARQL rewriting based on unified co-reference information. In: Proc. of ICDEW Workshops (2014)
21. Torre-Bastida, A.I., Bermúdez, J., Illarramendi, A., Mena, E., González, M.: Query rewriting for an incremental search in heterogeneous Linked Data sources. Flexible Query Answering Systems pp. 13–24 (2013)